



DECUS

PROGRAM LIBRARY

DECUS NO.	8-271
TITLE	LIP, <u>LOGICAL</u> " <u>IF</u> " <u>PACKAGE</u>
AUTHOR	Bryan D. Young
COMPANY	University Department of Medical Cardiology Royal Infirmary Glasgow, Scotland
DATE	February 17, 1970
SOURCE LANGUAGE	PALD

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

DECUS

LIBRARY



THE UNIVERSITY OF CHICAGO
LIBRARY
1100 EAST 58TH STREET
CHICAGO, ILL. 60637
TEL: 773-936-5000
FAX: 773-936-5001
WWW.CHICAGO.EDU

LIP, LOGICAL "IF" PACKAGE

DECUS Program Library Write-up

DECUS No. 8-271

ABSTRACT

LIP enables logical "IF" statements involving floating point variables to be written in a conventional manner within assembly language programs. The logical relations and operators available are:-

<u>Mnemonic</u>	<u>Meaning</u>
GT	Greater than
GE	Greater than or equal to
EQ	Equal to
LE	Less than or equal to
LT	Less than
NE	Not equal to
AN	Logical AND
OR	Logical OR

e.g. the FORTRAN statement:-

IF ((A.GT.B)OR(C.LT.D)) LOC

is written in machine code format as:-

IF;A;GT;B;OR;C;LT;D;GO;LOC where A,B,C and D are 3 word floating point variables.

3. Requirements

3.1 Storage.

This package occupies locations 4, 5400-5577. The 3 word Floating Point package (Digital 8-5-S) which must also be in core occupies locations 5-7, 40-63, 5600-7577.

Therefore total storage requirements are 4-7, 40-63,
5400-7577.

3.2 Equipment.

Standard PDP-8 , ASR-33 Teletype.

4. Usage

4.1 Loading.

This package is loaded with the binary loader (Digital 8-2-U).

The binary tape supplied contains only the LIP package and
hence the 3 word Floating Point package (Digital 8-5-S)
must also be loaded.

4.2 Calling Sequence.

Location 4 is used as an indirect address for entry to the
package. This is accomplished by use of the IF instruction
which is equivalent to JMS I 4 (see Discussion).

4.3 Definitions.

Since the mnemonic instructions used in the "IF" statements
are not held in the permanent symbol tables of the PAL and
MACRO compilers they must be defined by the user's
program as follows:-

IF	=	4404
GO	=	7777
OR	=	7776
AN	=	7775
GT	=	7774
GE	=	7773
EQ	=	7772
LE	=	7771
LT	=	7770
NE	=	7767

5. Restrictions

None.

6. Description

6.1 Discussion

LIP enables logical "IF" statements to be written within assembly language programs in a format similar in function and appearance to the corresponding FORTRAN and ALGOL statements. The general form is:-

IF;(logical expression);GO;LOC (1)

The complete statement is normally written on one line with semicolons between each variable name and instruction.

However, if a statement is too long it may be continued on subsequent lines.

Each statement must include both the IF and GO instructions which initiate the entry to and exit from the package respectively.

The logical expression, indicated in (1) is evaluated by the package and if it is found to be true control is transferred to the location named after the GO instruction (LOC in (1)). If it is not true control is returned to the next location after the end of the IF statement. The logical expression is composed of floating point variable names combined using the following relations and logical operators.

<u>Mnemonic</u>	<u>Meaning</u>
GT	Greater than
GE	Greater than or equal to
EQ	Equal to (=)
LE	Less than or equal to
LT	Less than
NE	Not equal to
AN	Logical AND operator
OR	Logical OR operator

Examples of logical expressions:-

A > B	A;GT;B
(A < B)OR(C > D)	A;LT;B;OR;C;GT;D
(A = B)AND (C = D)	A;EQ;B;AN;C;EQ;D
A > B > C	A;GT;B;AN;B;GT;C

All variable names must be 3 word floating point numbers.

Although there is no limit to the size of logical expressions, it is important to note that evaluation takes place from left to right.

e. g. A;GT;B;AN;C;LT;B;OR;C;LE;D is equivalent to:-

$((A > B) \text{ AND } (C < B)) \text{ OR } (C \leq D)$

6.2 Example.

Let A, B, C and D be floating point variables. It is required to know whether any of the following conditions hold:-

- (I) A=B=C=D
- (II) A > B > C > D
- (III) A < B < C < D

This would be programed as:-

```
IF;A;EQ;B;AN;B;EQ;C;AN;C;EQ;D;GO;L1
IF;A;GT;B;AN;B;GT;C;AN;C;GT;D;GO;L2
IF;A;LT;B;AN;B;LT;C;AN;C;LT;D;GO;L3
```


L4, / (I), (II), (III) are not true

L1, / (I) is true.

L2, / (II) is true.

L3, / (III) is true.

6.3 Errors.

Any format errors in an IF statement will cause the package to halt. Logical variables are not allowed and therefore a statement such as:- IF;A;GO;LOC is illegal and would produce a failure. The logical operators OR and AN may link only expressions of the type A;GT;B and hence the statement:-

IF;A;OR;B;GT;C;GO;LOC

is illegal. The correct version of this statement is:-

IF;A;GT;C;OR;B;GT;C;GO;LOC

A variable name may not be both preceded and followed by one of the relations GT, GE etc. Therefore the statement:-

IF;A;GT;B;GT;C;GO;LOC is illegal.

The correct version is:-

IF;A;GT;B;AN;B;GT;C;GO;LOC.

/LIP, A LOGICAL "IF" ROUTINE
/FOR ASSEMBLY LANGUAGE PROGRAMS

/DEFINITIONS:-

IF=4404
GO=7777
OR=7776
AN=7775
GT=7774
GE=7773
EQ=7772
LE=7771
LT=7770
NE=7767
FGET=5000
FSUB=2000
FEXT=0000

0004	5400	*4	5400	/INDIRECT ADDRESS FOR ROUTINE
5400	0000	*5400		/USED TO EXAMINE INSTRUCTIONS
5401	3237	POINTR,	0	
5402	5204		DCA CLV	
5403	2200	LP1,	JMP .+2	
5404	2200		ISZ POINTR	
5405	4240	LP4,	ISZ POINTR	
5406	7450		JMS FUNC	
5407	5226		SNA	
5410	7700		JMP OUT	
5411	5215		SMA CLA	
5412	4277		JMP LP2	
5413	3237		JMS VAL	
5414	5203		DCA CLV	
5415	2200	LP2,	JMP LP1	
5416	2200		ISZ POINTR	
5417	4240		ISZ POINTR	
5420	7700		JMS FUNC	
5421	7402		SMA CLA	
5422	4277		HLT	/ILLEGAL FORMAT
5423	4261		JMS VAL	
5424	3237		JMS LOG	
5425	5203		DCA CLV	
5426	2200	OUT,	JMP LP1	/EXIT FROM PACKAGE
5427	1237		ISZ POINTR	
5430	7650		TAD CLV	
5431	5235		SNA CLA	
5432	1600		JMP LP6	
5433	3200		TAD I POINTR	
5434	5600		DCA POINTR	
			JMP I POINTR	

5435	2200	LP6,	ISZ POINTR	
5436	5600		JMP I POINTR	
5437	0000	CLV,	Ø	/CURRENT LOGICAL VALUE
5440	0000	FUNC,	Ø	/DETERMINES INSTRUCTIONS
5441	1600		TAD I POINTR	
5442	7500		SMA	
5443	7402		HLT	/ILLEGAL INSTRUCTION
5444	7001		IAC	
5445	7450		SNA	
5446	5640		JMP I FUNC	
5447	1377		TAD (2)	
5450	7510		SPA	
5451	5254		JMP .+3	
5452	7201		CLA IAC	
5453	5640		JMP I FUNC	
5454	1376		TAD (6)	
5455	7710		SPA CLA	
5456	7402		HLT	/ILLEGAL INSTRUCTION
5457	7040		CMA	
5460	5640		JMP I FUNC	
5461	0000	LOG,	Ø	/EVALUATES 'AND', 'OR'
5462	1237		TAD CLV	
5463	3275		DCA RHS	
5464	1200		TAD POINTR	
5465	1375		TAD (-2)	
5466	3276		DCA LTEM	
5467	1676		TAD I LTEM	
5470	7001		IAC	
5471	1275		TAD RHS	
5472	7700		SMA CLA	
5473	7001		IAC	
5474	5661		JMP I LOG	
5475	0000	RHS,	Ø	
5476	0000	LTEM,	Ø	
5477	0000	VAL,	Ø	/EVALUATES 'A;GT;B' ETC.
5500	1200		TAD POINTR	
5501	1374		TAD (-1)	
5502	3352		DCA VAL2	
5503	1752		TAD I VAL2	
5504	3351		DCA VAL1	
5505	2352		ISZ VAL2	
5506	2352		ISZ VAL2	
5507	1752		TAD I VAL2	
5510	3352		DCA VAL2	
5511	4407		JMS I 7	
5512	5751		FGET I VAL1	
5513	2752		FSUB I VAL2	
5514	0000		FEXT	

5515	1045		TAD 45
5516	7450		SNA
5517	5327		JMP EQUAL
5520	7700		SMA CLA
5521	5337		JMP GRET
5522	1600		TAD I POINTR
5523	1376		TAD (6)
5524	7710		SPA CLA
5525	7001		IAC
5526	5677		JMP I VAL
5527	1600	EQUAL,	TAD I POINTR
5530	1373		TAD (4)
5531	7500		SMA
5532	5677		JMP I VAL
5533	1372		TAD (3)
5534	7700		SMA CLA
5535	7001		IAC
5536	5677		JMP I VAL
5537	1600	GRET,	TAD I POINTR
5540	1371		TAD (5)
5541	7510		SPA
5542	5345		JMP .+3
5543	7201		CLA IAC
5544	5677		JMP I VAL
5545	1372		TAD (3)
5546	7710		SPA CLA
5547	7001		IAC
5550	5677		JMP I VAL
5551	0000	VAL1,	0
5552	0000	VAL2,	0
5571	0005		
5572	0003		
5573	0004		
5574	7777		
5575	7776		
5576	0006		
5577	0002		

AN	7775
CLV	5437
EQ	7772
EQUAL	5527
FEXT	0000
FGET	5000
FSUB	2000
FUNC	5440
GE	7773
GO	7777
GRET	5537
GT	7774
IF	4404
LE	7771
LOG	5461
LP1	5403
LP2	5415
LP4	5405
LP6	5435
LT	7770
LTEM	5476
NE	7767
OR	7776
OUT	5426
POINTR	5400
RHS	5476
VAL	5477
VAL1	5551
VAL2	5552

